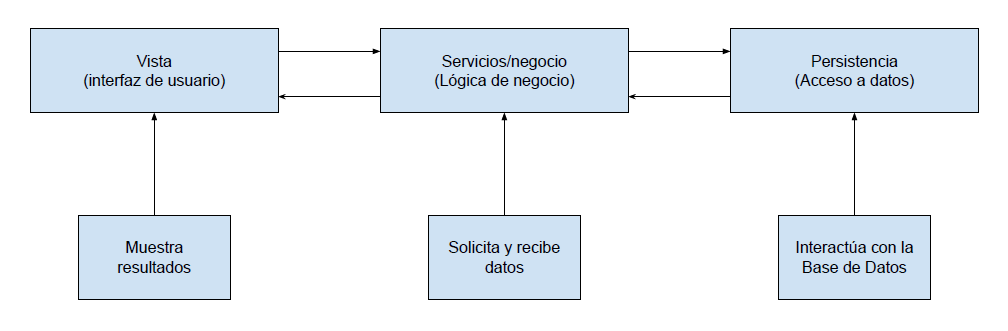
## **Arquitectura en Capas en Java**



Una arquitectura en capas separa la aplicación en módulos lógicos, donde cada módulo tiene una responsabilidad específica y bien definida. Esto mejora la **modularidad**, la **mantenibilidad** y la **escalabilidad** de la aplicación.

### **1. Capa de Persistencia (Data Access Layer - DAL)**

Esta capa es responsable de la comunicación con la base de datos. Su función principal es traducir los objetos de Java a registros de la base de datos y viceversa. No debería contener lógica de negocio.

**Componentes clave:**

* **DAO (Data Access Object):** Son interfaces y sus implementaciones que definen las operaciones CRUD (Crear, Leer, Actualizar, Borrar) para una entidad específica.

### **2. Capa de Negocio / Servicios (Business Logic Layer / Service Layer)**

Esta es la capa central de la aplicación. Contiene toda la **lógica de negocio** y coordina las operaciones entre la capa de persistencia y la capa de presentación. Los servicios suelen invocar uno o más DAOs para realizar sus funciones.

**Componentes clave:**

* **Servicios:** Clases que encapsulan la lógica de negocio. Realizan operaciones complejas que pueden involucrar múltiples interacciones con la base de datos o validaciones.

### **3. Capa de Vista (Presentation Layer / User Interface)**

Esta capa es responsable de la interacción con el usuario. Muestra la información y captura las entradas del usuario. Delega las operaciones de negocio a la capa de servicios. No debe contener lógica de negocio ni interactuar directamente con la base de datos.

**Componentes clave:**

* **Interfaces de usuario:** Pueden ser aplicaciones de escritorio (Swing, JavaFX), aplicaciones web (Servlets, JSP, Spring MVC, JSF, Thymeleaf, React, Angular), o APIs RESTful.